The Problem: Incorporating Identifiers Into a Data Server

Anne Wilson
Laboratory for Atmospheric and Space Physics
University of Colorado, Boulder
anne.wilson@lasp.colorado.edu

ESDSWG Annual Meeting
October 22, 2010
New Orleans

The Setting:
LaTiS Data Server

- Earlier version called TSDS
- Targets data access
- Serves times series data: data as a function of time
- Reads a variety of input formats into a common data model
  - Can serve binaries, data from a RDMBS
- Serves a variety of output formats
- Can serve remote datasets
- Supports subsetting on time, variables
- Supports filtering, e.g. thinning
- Will support transformations, aggregation

Goals, Tasks

- Goal: Support data identification, citation, and provenance, "scientific equivalence"
- Task: Assign and provide DOIs for a dataset, UUID for granules
  - Put them in the metadata
  - For our datasets that are self describing, put them in the granules

Issue: for non self describing formats, such as binary, clients would need to make a separate call to get the identifiers

- We must maintain these mappings in perpetuity
  - Scary

Issue: what to name them?

- “DOI”, “UUID”
- What will other providers call them?

Issue: What Gets a unique ID?

- A new dataset. Triv? Not!
- What is a new dataset? We decided:
  - A different output format **is not** a new dataset, keep original IDs
    - Maintains scientific equivalence
    - Issue: it is important for users to understand that reformatting the data can have consequences that impact data usage and analysis, e.g., numeric representation issues
    - Dataset is tightly coupled to server
  - A subset based on time or variable **is not** a new datasets, keep original IDs
  - A continuously updated dataset **is not** a new dataset
  - A filtered dataset, e.g., a thinned dataset **is not** a new dataset
  - A reprocessed dataset, e.g., a new version, **is** a new dataset, needs new IDs
  - Transformations, aggregations **are** new datasets generated on demand, but are ephemeral, not maintained by our server
    - Leave responsibility of creating, assigning any IDs to data creators, clients of our server. Hope they cite us.
  - What will other servers do?

Issue: Verification: How to prove that a replica of a dataset is complete, intact, and a copy of the original file?

- We can claim scientific equivalence, but proving it gets murky
- Each of our datasets has a "reference representation", the form in which it is stored on the disk
- Use of a checksum won't work to show a proper subset is the same dataset as its source
- Possible solution: user resends the original request that created the dataset but requests a checksum (instead of data) that could be used for comparison.
  - Here the original request is serving as yet another identifier.
  - Tight coupling to server
  - Identification of the service itself?
  - Problem: if client can't determine original request than proof can't be made

Issue: What does "publishing" a dataset mean?

- Making data available on the web?
- Giving a copy to a user via FTP?
- Registering with a naming authority, such as DataCite?
- Being referenced in a published paper?
- We decided publishing means web availability, registration
  - These are deliberate actions we can take, as opposed to waiting (and hoping) for a reference in a published paper to appear
- Pledging to maintain dataset IDs in perpetuity is required
  - We will not maintain metadata for dynamically generated datasets, even though we provide access via the web
- If we serve a remote dataset are we publishing it?

Issue: What about remote datasets served by LaTiS?

- We maintain some minimal metadata for those datasets
- But we are not their owners so will not generate IDs for them
    - Owners may not have IDs for their datasets, but that is beyond the scope of our responsibility
    - If an ID is available, we should be able to provide it
- Will other data providers do the same?

Issue: Can we ensure that we get credit for datasets created by LaTiS?

- If a client creates and publishes work based on a new dataset made using our server we would like to be cited
- We can't enforce citation, we can only hope users do the right thing and cite us
- Is this the best we can do? Perhaps. No worse than current situation.

Conclusion: Assigning IDs to datasets and granules is easy technically but has hard cultural issues

- The community must establish and follow practices regarding
  - who assigns IDs
  - what should have an ID and what shouldn't
  - how IDs are propagated
- Data server can be tightly coupled to datasets served
  - Different implementations of same protocols could produce different results
  - Services themselves may need formal identification and warrant separate citation